# Combining Acoustic Feature Sets for Detecting Mild Cognitive Impairment in the Interspeech'24 TAUKADIAL Challenge

Gábor Gosztolya[1,2], László Tóth[1]

[1] University of Szeged, Institute of Informatics, Szeged, Hungary
[2] ELRN-SZTE Research Group on Artificial Intelligence, Szeged, Hungary

## 1. The Focus of This Study

- We present the approach of our team for MCI detection in the course of the Interspeech'24 TAUKADIAL Challenge
- We focus entirely on acoustics, employing four feature types

### The MCI Recordings

- 169 English & Chinese subjects (train: 129, test: 40)
- Three recordings for each subject (picture description)
- Classification task: MCI vs. HC (evaluated by UAR — we used $F_1$...)
- Regression task: estimating the MMSE score (evaluated by RMSE)
- Up to five predictions could be submitted at the same time

## 3. Experimental Setup and Meta-Parameter Setting

### Pre-processing

- Re-sampling the utterances to 16 kHz mono
- Automatic volume normalization

### Classification, Meta-Parameter Setting

- Support Vector Machines (SVM) + linear kernel
- 20-fold cross-validation, repeated five times (with different speakers)
- Performance metrics were averaged out, and the $C$ value with the best mean score was chosen
- Final classifier / regression model was trained on all the recordings of the training set with this optimal $C$ value

### Prediction Fusion

- We treated the three picture description recordings as separate tasks, but fused the predictions of the SVM / SVR models for the subjects
- We took the unweighted mean of the posteriors / MMSE predictions
- Fusing feature sets was done in the same way (unweighted mean)

### Utterance Chunking

- 30s long chunks with 50% overlap (minimal duration: 10s)
- Classification was still done on the subject level
- Predictions were also merged to subject level via unweighted mean

## 3. Feature Extraction

### ComParE functionals

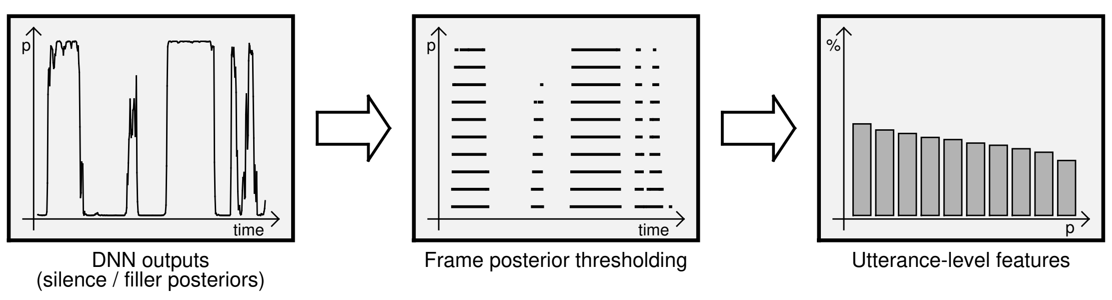
- A standard general feature set, based on frame-level attributes and their statistics (e.g. mean, standard deviation, peak statistics)
- 6737 attributes, calculated by the python port of openSMILE

### Pause statistics

This method describes the amount of pause present in the recording

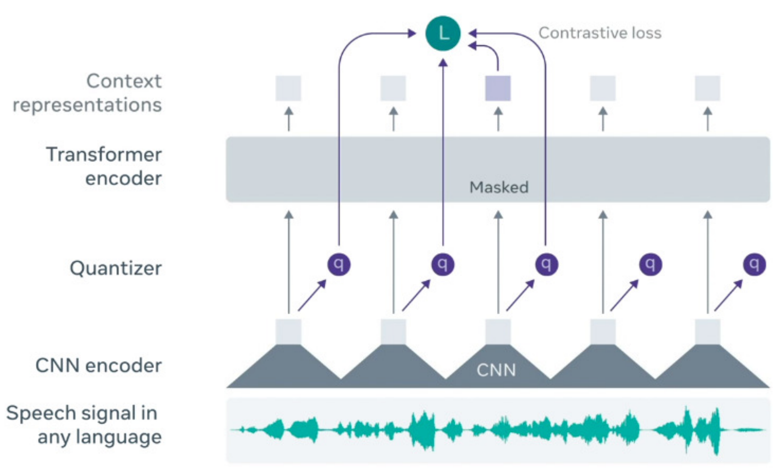
(1) A standard HMM/DNN acoustic model is evaluated
(2) The local probabilities of **silent** and **filled pauses** are calculated
(3) The ratio of frames where this probability exceeds a threshold is noted

It is repeated with thresholds between 0 and 1 with 0.02 increments.




### wav2vec 2.0 Embeddings

- We use a large wav2vec 2.0 XLSR-53 model, fine-tuned on 2182 hours (English)
- Embeddings were taken from the last layers of the **convolutional** and contextualized (**fine-tuned**) blocks
- Frame-level embeddings were aggregated via mean and standard deviation




This study was supported by the NRDI Office of the Hungarian Ministry of Innovation and Technology (grant TKP2021-NVA-09), and within the framework of the Artificial Intelligence National Laboratory Program (RRF-2.3.1-21-2022-00004).

## 5. Cross-Validation Results for Individual Features

We report the results aggregated on the subject level (i.e. not individually for the specific speech tasks).
Cells in dark gray refer to approaches (later) evaluated on the test set.

| Feature set | Chunks | Classification $F_1$ | Classification AUC | Regression Corr. | Regression RMSE |
|---|---|---|---|---|---|
| ComParE functionals | — | 71.52% | 0.677 | 0.455 | 2.971 |
| | yes | 72.73% | 0.668 | 0.404 | 3.079 |
| Pause statistics | — | 61.87% | 0.605 | 0.525 | 2.887 |
| | yes | 54.84% | 0.643 | 0.489 | 2.951 |
| wav2vec 2.0 (Conv.) | — | 73.55% | 0.721 | 0.421 | 3.054 |
| | yes | 73.20% | 0.698 | 0.444 | 2.988 |
| wav2vec 2.0 (Fine-tuned) | — | 73.08% | 0.655 | 0.533 | 2.833 |
| | yes | 70.20% | 0.681 | 0.479 | 2.932 |

- Chunking was usually not effective (we omit it from the further tests)
- For classification, Pause statistics was not really good
- Again, for classification, convolutional wav2vec 2.0 embeddings outperformed the fine-tuned ones
- For regression, it was the other way around: Pause statistics and the fine-tuned wav2vec 2.0 embeddings were the best features

## 6. Cross-Validation Results for Feature Set Combinations

We used the Sequential Forward Feature (set) Selection method

(1) First we took the feature set with the best performance
(2) We tried adding each remaining feature set to the already selected feature set combination
(3) We chose the best variation, then repeated step (2)

### Results for Classification

| Feature sets | $F_1$ | AUC |
|---|---|---|
| wav2vec 2.0 (Conv.) + ComParE functionals | 76.43 | 0.726 |
| wav2vec 2.0 (Conv.) + Pause stats | 70.59 | 0.710 |
| wav2vec 2.0 (Conv.) + wav2vec 2.0 (Fine-tuned) | 76.25 | 0.733 |
| wav2vec 2.0 (Conv.) + ComParE func. + wav2vec 2.0 (Fine-tuned) | 75.31 | 0.729 |
| wav2vec 2.0 (Conv.) + ComParE func. + Pause stats | 75.32 | 0.720 |
| All four methods | 75.16 | 0.725 |

- Fusing the convolutional embeddings and the ComParE functionals (or the fine-tuned embeddings) was beneficial
- Three-wise and four-wise combinations were futile

### Results for Regression

| Feature sets | $F_1$ | AUC |
|---|---|---|
| wav2vec 2.0 (Fine-tuned) + ComParE functionals | 0.532 | 2.839 |
| wav2vec 2.0 (Fine-tuned) + Pause stats | 0.597 | 2.769 |
| wav2vec 2.0 (Fine-tuned) + wav2vec 2.0 (Conv.) | 0.514 | 2.914 |
| wav2vec 2.0 (Fine-tuned) + Pause stats + ComParE func. | 0.585 | 2.790 |
| wav2vec 2.0 (Fine-tuned) + Pause stats + wav2vec 2.0 (Conv.) | 0.565 | 2.839 |
| All four methods | 0.586 | 2.812 |

- Fusing the fine-tuned embeddings and the Pause statistics was beneficial, but the other binary combinations made the results worse
- Three-wise and four-wise combinations were futile, again

## 7. Test Set Results

### Results for Classification

| Feature sets | $F_1$ | UAR |
|---|---|---|
| ComParE functionals | 52.2 | 44.4% |
| wav2vec 2.0 (Conv.) | 51.2 | 47.2% |
| wav2vec 2.0 (Conv.) + ComParE functionals | 56.5 | 49.4% |
| wav2vec 2.0 (Conv.) + wav2vec 2.0 (Fine-tuned) | 56.5 | 49.4% |
| wav2vec 2.0 (Conv.) + ComParE func. + wav2vec 2.0 (Fine-tuned) | 52.2 | 44.1% |

- All values are below chance level (UAR ¡ 50%)...

### Results for Regression

| Feature sets | Corr. | RMSE |
|---|---|---|
| Pause statistics | 0.439 | 2.608 |
| wav2vec 2.0 (Fine-tuned) | 0.457 | 2.660 |
| wav2vec 2.0 (Fine-tuned) + Pause stats | 0.455 | 2.612 |
| wav2vec 2.0 (Fine-tuned) + Pause stats + ComParE func. | 0.400 | 2.702 |
| All four methods | 0.407 | 2.683 |

- The RMSE values were quite similar to those in cross-validation
- This robustness validates our meta-parameter setting procedure
- All our submissions were better than the baseline RMSE value (2.89)

## 7. Conclusions

- The really bad results for classification might be due to optimizing for the wrong metric ($F_1$ instead of UAR)
- ...or it can be attributed to the given dataset (the baseline results are not convincing either, lacking any robustness)
- However, the regression scores were competitive on the test set